Medical Information Extraction from Unstructured Clinical Text

Jason Khuu
Electrial & Computer Engineering
University of Toronto
jason.khuu@mail.utoronto.ca

Weihan Xie
Electrial & Computer Engineering
University of Toronto
weihan.xie@mail.utoronto.ca

Abstract

Textual medical records can be long and time consuming to review, resulting in less time medical professionals can spend offering patient care. The objective of this project was to use a discriminative model to highlight clinically important information using Named Entity Recognition. The model architecture selected was a BiLSTM-CRF, this supervised learning problem required an LLM to initially generate the "true" labels since the initial dataset did not contain any and after further data preprocessing, was used to train the model. After applying regularization and early stop to combat overfitting, the final validation accuracy was 90.95% and the final test accuracy was 90.90%.

Assentation of Teamwork

Jason indepdently completed the model design and dataset research, wrote the report, code, presentation slides and did the presentation. Weihan contributed early ideation for project topics.

1 Introduction

Electronic Health Records contains a vast repository of patient information in both structured and unstructured formats. Extracting key medical information such as problems, treatments and anatomic locations is required to make decisions for patient care. However, much of this information is disjointed and buried within unstructured clinical text. Therefore, using techniques such as Named Entity Recognition (NER) to extract this information can greatly increase the efficiency for health care providers to access this relevant information from EHRs.

While trained large language models (LLMs) can perform such NER task, they are not ideal in a medical setting given their randomness and inconsistencies in output. Therefore, the goal of this project was to implement the discriminative model known as bi-directional long short-term memory (BiLSTM) with a conditional random field (CRF), BiLSTM-CRF to solve the problem.

2 Preliminaries and Problem Formulation

The end goal was to implement an NER model that can assign a BIO (Beginning, Inside, Outside) label for problems, treatments, anatomic locations, measurements and test results to unlabeled medical text. This means the tokenized input sequence of text, where each x_i is a word in the sequence:

$$X = [x_1, x_2, \ldots, x_T], \quad Y = [y_1, y_2, \ldots, y_T]$$

Must be converted into the respective output label sequence, where each y_i is a label from the set of possible BIO labels. An expected output example is seen in Figure 1.

ECE 1508: Applied Deep Learning (Fall 2025).

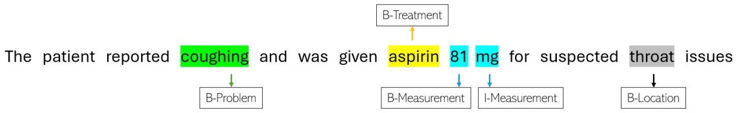
B-Treatment
The patient reported coughing and was given aspirin 81 mg for suspected throat issues
B-Problem
B-Measurement
I-Measurement
B-Location

Figure 1: BiLSTM-CRF model output expectation

3 Solution via Deep Learning

To design this NER model, the synthetic medical text dataset SimSUM will be used. This dataset was selected because real EHR data can often be prone to bias from the medical professional it was authored by. The goal of SimSUM was to help over come this barrier by creating a more diverse and less bias dataset that could be used for research projects interested in medical text, by generating samples based on probabilistic models and LLMs. A data sample from SimSUM contains:

- **Tabular Data:** Tabular representation of patient's medical history
- **Natural Text:** Clinical text report in natural language

The first layer of the architecture will be used to convert the tokenized text sequence into its embedded representation. The output of the embedding model for input sequence X is:

$$M = [m_1, m_2, \ldots, m_T]$$

Where each m_t is the word embedding vector for each respective token in the input sequence. This tensor can then be passed to the BiLSTM to compute the contextual relavence of each token both forwards and backwards through the sequence. A BiLSTM architecturally contains two LSTMs with opposite start and end points. The final output of the BiLSTM is simply the two individual LSTM output vectors merged in some form, for this model the merging method selected was concatenation.

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}], \quad H = [h_1, h_2, \ldots, h_T]$$

Where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are the hidden states from the forward and backward LSTM respectively. Using this context aware vector the emissions scores can be calculated through a simple linear layer where the output dimension is equal to the number of possible labels with a score for each.

$$e_t = [label_1, label_2, \ldots, label_N], \quad E = [e_1, e_2, \ldots, e_T]$$

The final CRF layer of the model is then used to calculate the transition scores using the transfer matrix A which consist of a set of learnable weights and this score is used to calculate the final output Y which is output probability of BIO labeling for each element in the sequence.

$$\text{Score}(x, y) = \sum_{t=1}^{T} E_t + \sum_{t=1}^{T} A_{y_{t-1}, y_t}, \quad P(y \mid X) = \frac{\exp(\text{Score}(x, y))}{\sum_{\tilde{y} \in \mathcal{Y}(X)} \exp(\text{Score}(x, \tilde{y}))},$$

The loss function in this case is equal to the negative log likelihood $L = -\log P(y \mid X)$ and was minimized using the known correct label sequence given at the input as part of the training set. Afterwards, hyperparameter were tuned using the validation data and final assessment made using the test data. Since label data is not provided within SimSUM, an LLM agent wass required to intially help with generating the dataset labels for training.

4 Implementation

4.1 Data Labelling

The main library used for data annotation was the Ollama REST API, which enabled experimentation with different local LLMs and prompt engineering techniques to generate the BIO labels. As illustrated in Figure 2, the process begins by sending a single data sample's raw text and associated metadata to an LLM agent. Using the system prompt mentioned in the appendix, the the agent generated the labelled text in the following format: "`word<label>`, `word<label>`, ..."

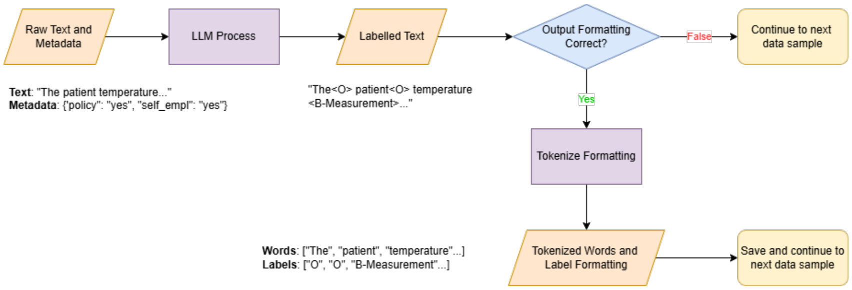
Raw Text and Metadata
LLM Process
Labelled Text
Output Formatting Correct?
False
Continue to next data sample
Text: "The patient temperature..."
Metadata: {'policy": "yes", "self_empl": "yes"}
"The<O> patient<O> temperature <B-Measurement>..."
Yes
Tokenize Formatting
Words: ["The", "patient", "temperature"...]
Labels: ["O", "O", "B-Measurement"...]
Tokenized Words and Label Formatting
Save and continue to next data sample

Figure 2: Data Labelling Flowchart for One Sample

Given the potential for hallucination when using the LLM for labelling, this text output is filtered using basic string logic to determine if the formatting is correct. Successful samples would have the words and labels extracted, tokenized and saved before proceeding to the next instances, while cases that fail this check would be deleted. For this project, the agent was limited to output 11 predefined BIO labels, all are captured within the system prompt mentioned in the appendix.

4.2 Data Preprocessing

Before training the data must be convereted into a format understandable by the model, this flow is illustrated in Figure 3. The words and labels in a tokenized state were converted into a integer ID representation, this was be done through creating a look up table for words and labels in the train set. Specifically, when making the work lookup table two special tokens were included:

- `[PAD]` Used to extend all sentences in the same batch to be the same length
- `[UKN]` Used to tokenize unknown words not within the vocabulary

The main motivation for `[PAD]` is to fill parallel computational requirements for batch training while letting the model know real from padded data. However, `[UKN]` contributes to how well the model can generalize to words outside of its training material. The hyperparameter `minimum frequency` is used to decide the number of times a word must appear to be considered part of the vocabulary. If this is set to 1 this means all training data words will be in the vocabulary, meaning `[UKN]` is never used in training and the model will not know how to act with unknown data. Tuning for this will be explored in the numerical experimentation section.

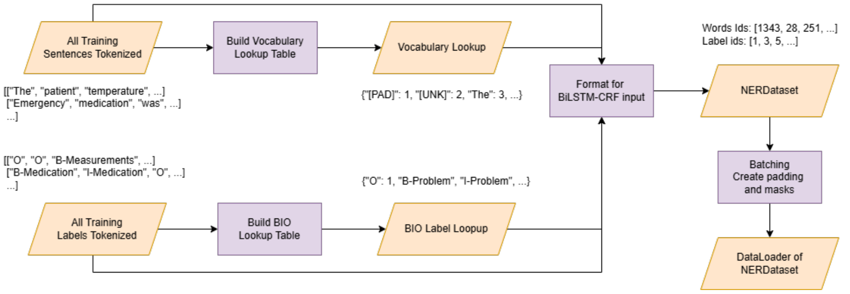
All Training Sentences Tokenized
Build Vocabulary Lookup Table
Vocabulary Lookup
Words Ids: [1343, 28, 251, ...]
Label ids: [1, 3, 5, ...]
[["The", "patient", "temperature", ...]
["Emergency", "medication", "was", ...]
...]
{"[PAD]": 1, "[UNK]": 2, "The": 3, ...}
Format for BiLSTM-CRF input
NERDataset
[["O", "O", "B-Measurements", ...]
["B-Medication", "I-Medication", "O", ...]
...]
{"O": 1, "B-Problem", "I-Problem", ...}
Batching Create padding and masks
All Training Labels Tokenized
Build BIO Lookup Table
BIO Label Loopup
DataLoader of NERDataset

Figure 3: Data preprocessing illutration required before training and inference

4.3 Model Design

The implementation of the BiLSTM-CRF model was completed using PyTorch and the architecture is visualized in Figure 4. There are two paths mentioned within this diagram; training and inference but both paths are identical up to the emission score calcucation. The input data is from `NERDataset` that was computed based on section 4.2, the tokenized word IDs (of sequence length `T`) are passed to an embedding layer that converts each ID into a vector word embedding (of dimension `E` = 128) and then some elements are masked in the following dropout layer. This tensor is then inserted into the

BiLSTM model which calculates the contextual relavence between each embedding from start to end ($\overrightarrow{h_t}$), end to start ($\overleftarrow{h_t}$) and contatenated together to form the the hidden layer output h_t (of dimension H = 256). The next dropout layer then mask some of these values and get passed to the linear layer to calculate the emissions scores of each token in the sequence for every possible label.

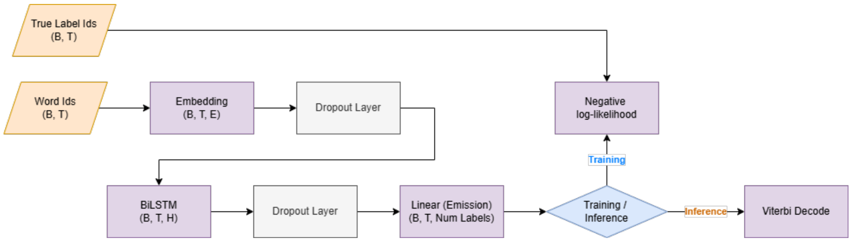

Figure 4: BiLSTM-CRF training and inference illustration

If the model was being trained, then the negative log-likelihood was computed using the true label IDs and weights were updated through backwards propagation. However, if it was used for inference then the CRF to use Viterbi's algorithm on the emissions scores to calculate the final labels. The train data was used to update the weights, and the validation set was used to tune parameters and adjust the design. After a final validation loss curve was settled on the final evaluation was done using the test set and the results were recorded.

5 Numerical Experiments

5.1 Data Labelling

When labelling the data with the LLMs it was identified that some models would noticeably hallucinate the instructions more often than others. As well, some models would generate a response much slower than others, which is significant in this case since 10,000 sentences needed to be processed. Three models were assessed to see which could obtain the best results given these criteria and results were recorded in Table 1.

	Llama 3.1 8B	**Qwen 3 8B**	**Gemma 3 12B**
Time to Process One Sample (seconds)	4.83	28.47	10.17
Formatting Successes Percentage (%)	5	50	65

Table 1: Model Performance Comparison

In the end Llama 3.1 8B provided the quickest responses but it's formatting success was very rare. Gemma 3 12B on in the end had the highest formatting success rate of 65% and took on average 10 seconds for each sentence making it the best choice. After processing 10,000 samples, 7,215 samples were formatted successfully. Although more prompt engineering experimentation could have been conducted to increase this number it was a sufficient for train this project.

5.2 Data Preprocessing

The dataset that was successfully labelled was divided into a training set (70%), validation set (15%) and test set (15%). Using the training set, the vocabulary was created with various `minimum frequency` values and experiments were recorded in Table 2. To know how to interrupt these results consider, when `minimum frequency = 1`, the remaining words in the training set was 100% because all the words were included in the vocabulary, this resulted in the vocabulary being a length of 7003. Then when comparing the hypothetical vocabulary of the validation set with this, 87.8% of the data was the same. This means the words within that remaining 12.2% will be set as `[UKN]` during the embedding process and without any `[UKN]` in the training material then the model will not know how to label these well.

By setting `minimum frequency = 5`, it could be seen that many of the words in the vocabulary could be viewed as "noise" to a dataset, because only 0.7% of the tokens were removed but this translated to 57.1% of the vocabulary being eliminated. Based on the third column, it could be seen that the validation data will not have more words missing from the train data vocabulary but intuitively can already see that the driving factors for this reduction was that the noise in validation data is no longer in the training set. `minimum frequency = 10` was also tested but for this project using 5 was selected to avoid shrinking the vocabulary length by too much. This helps ensure that during training, some `[UKN]` tokens were included in the dataset while not excluding useful information.

Minimum	**Remaining Words (%)**	**Vocab Length**	**| Validation ∩ Training |**
Frequency	(In Train Data)	(In Train Data)	**/ | Validation | (%)**
1	100	7003	87.8%
5	99.3	3014	66.5%
10	98.8	2248	53.3%

Table 2: Minimum frequency impact on training data vocabulary

5.3 Model Results

The "default model" shown in Figure 5, contains no regularization or unique training techniques it is simply uses `minimum frequency = 1`, `batch size = 64`, `embedded dimension = 128`, `LSTM dimension = 256` and `learning rate = 0.001`. It was visible from this graph that the model was overfitting to the train data, which meant no parameter search was needed and some form of regularization was required, this was when L1 regularization and dropout layer were added. The results were captured in Figure 6, using `lambda = 1e-05` and `dropout = 0.2`.

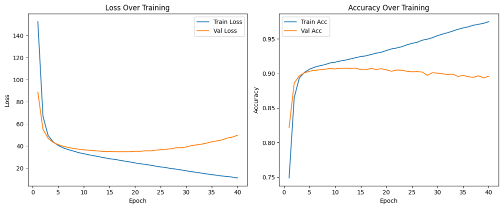

Figure 5: Default model (Val. Acc. = 89.60%)

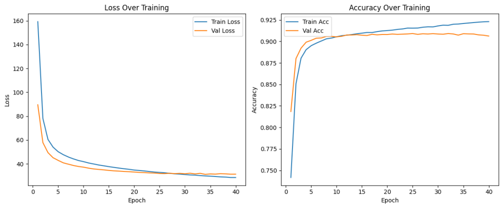

Figure 6: Regularization (Val. Acc. = 90.62%)

Afterwards, `minimum frequency = 5` was used and results were shown in Figure 7, but little impact was noticed. The final main observation was that the train loss would continue to decrease and accuracy would increase while the validation results stopped change. Therefore, to again minimize this overfitting early stop was implemented with hyperparameters `patience = 3`, `validation delta = 0.3` and `train delta = 0.1` and results were captured in Figure 8.

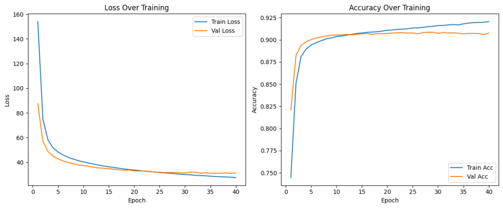

Figure 7: Min frequency (Val. Acc. = 90.75%)

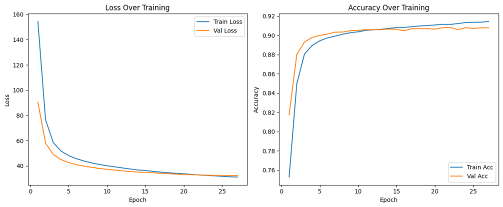

Figure 8: Early Stop (Val. Accuracy. = 90.95%)

Using the model from Figure 8, the test dataset was assessed and the final test accuracy was 90.90%.

6 Conclusions

In conclusion, BiLSTM-CRF was effective at performing NER task on clinical data achieving 90.90% test accuracy. As a further improvement in the future, using random masking for `[UKN]` across the token dataset could be used to see if it could increase generalization in a similar way dropout benefits model training as well. As well, the dataset used for this project was entirely based on a fake medical dataset that was labelled by an LLM, it was also using simplified note syntax instead of doctor shorthand. Therefore, to take this model to a level of application readiness, would require additional experimentation on more complex and real datasets as a next step.

References

[1] Huang, Z. & Xu, W. & Yu, K. (2015) *Bidirectional LSTM-CRF models for sequence tagging.* arXiv preprint arXiv:1508.01991.

[2] Rabaey, P. & Arno, H. & Heytens, S. & Demeester, T. (2024) *SynSUM–Synthetic Benchmark with Structured and Unstructured Medical Records.* arXiv preprint arXiv:2409.08936.

[3] Wen, S. & Xu, J. (2024) *Enhancing Time Series Prediction in Industrial Processes: A Self-Attention BiLSTM Approach.* In 2024 7th International Conference on Robotics, Control and Automation Engineering (RCAE), pp. 615–620. doi:10.1109/RCAE62637.2024.10834197.

Appendix

The system prompt used to label data is visible below:

```
You are a strict Named Entity Recognition (NER) tagger.

Your job is to assign labels using BIO (Beginning, Inside, Outside) tagging scheme
    to words in clinical text. The entity types you should be searching for a
    PROBLEM (example syymptoms, diseases), TREATMENT (example medications,
    procedures), LOCATION (example body parts).

The only valid labels you should ever assign are: B-PROBLEM, I-PROBLEM, B-TREATMENT,
     I-TREATMENT, B-LOCATION, I-LOCATION, B-MEASUREMENT, I-MEASUREMENT, B-TEST, I-
    TEST and O, where O means the token is not part of any entity.

To help you with the labelling process, the input format you will always be given is
     as follows:

Text: <clinial text to be labelled>
Metadata: <JSON dict of metadata>

The metadata should not be labelled at all it is only used as a support to help you
    underestand how to label the text. The exact format you should always return is:


Assuming the clinical text is "Text: Patient reports a significant increase in body
    temperature over the last 48 hours, exceeding normal ranges, indicating a high
    fever. There have been no respiratory symptoms such as pain, dyspnea, or cough.
     The patient illustrates general malaise and mentions feeling very fatigued due
     to the fever. No notable changes in daily routine or exposure to environments
    that might typically contribute to fever are reported. Recent stress levels and
     potential exposure to infectious agents during travels are also discussed.
    Vital signs show elevated temperature (103 °F). Heart rate is slightly
    tachycardic at 98 bpm, corresponding with the fever. Oxygen saturation is
    within normal limits at 98%, and lungs are clear to auscultation without any
    added sounds. Abdominal examination is normal, without tenderness or
    organomegaly. Skin shows no rashes, warmth, or lesions. Capillary refill time
    is adequate. Neurological assessment is non-focal. Overall, there are no
    evident physical findings to explain the fever apart from the stated
    temperature elevation. Metadata: {'policy': 'yes', 'self_empl': 'no', 'asthma':
     'no', 'smoking': 'no', 'COPD': 'no', 'season': 'summer', 'hay_fever': 'no', '
    pneu': 'no', 'common_cold': 'no', 'dysp': 'no', 'cough': 'no', 'pain': 'no', '
    fever': 'high', 'nasal': 'no', 'antibiotics': 'yes', 'days_at_home': np.int64(2)
    }" then you should just return

Patient<O> reports<O> a<O> significant<O> increase<O> in<O> body<B-MEASUREMENT>
    temperature<I-MEASUREMENT> over<O> the<O> last<O> 48<O> hours<O> ,<O> exceeding<
    O> normal<O> ranges<O> ,<O> indicating<O> a<O> high<B-PROBLEM> fever<I-PROBLEM>
     .<O> There<O> have<O> been<O> no<O> respiratory<B-PROBLEM> symptoms<I-PROBLEM>
     such<O> as<O> pain<B-PROBLEM> ,<O> dyspnea<B-PROBLEM> ,<O> or<O> cough<B-
    PROBLEM> .<O> The<O> patient<O> illustrates<O> general<B-PROBLEM> malaise<I-
    PROBLEM> and<O> mentions<O> feeling<O> very<O> fatigued<B-PROBLEM> due<O> to<O>
```

```
     the<O> fever<B-PROBLEM> .<O> No<O> notable<O> changes<O> in<O> daily<O>
    routine<O> or<O> exposure<O> to<O> environments<O> that<O> might<O> typically<O>
     contribute<O> to<O> fever<B-PROBLEM> are<O> reported<O> .<O> Recent<O> stress<
    O> levels<O> and<O> potential<O> exposure<O> to<O> infectious<B-PROBLEM> agents<
    I-PROBLEM> during<O> travels<O> are<O> also<O> discussed<O> .<O> Vital<O> signs<
    O> show<O> elevated<B-MEASUREMENT> temperature<I-MEASUREMENT> (<O> 103<B-
    MEASUREMENT> °F<I-MEASUREMENT> )<O> .<O> Heart<B-MEASUREMENT> rate<I-
    MEASUREMENT> is<O> slightly<O> tachycardic<B-PROBLEM> at<O> 98<B-MEASUREMENT>
    bpm<I-MEASUREMENT> ,<O> corresponding<O> with<O> the<O> fever<B-PROBLEM> .<O>
    Oxygen<B-MEASUREMENT> saturation<I-MEASUREMENT> is<O> within<O> normal<O>
    limits<O> at<O> 98<B-MEASUREMENT> %<I-MEASUREMENT> ,<O> and<O> lungs<B-LOCATION>
     are<O> clear<O> to<O> auscultation<O> without<O> any<O> added<O> sounds<O> .<O>

Abdominal<B-LOCATION> examination<B-TEST> is<O> normal<O> ,<O> without<O> tenderness
    <B-PROBLEM> or<O> organomegaly<B-PROBLEM> .<O> Skin<B-LOCATION> shows<O> no<O>
    rashes<B-PROBLEM> ,<O> warmth<B-PROBLEM> ,<O> or<O> lesions<B-PROBLEM> .<O>
    Capillary<B-MEASUREMENT> refill<I-MEASUREMENT> time<I-MEASUREMENT> is<O>
    adequate<O> .<O> Neurological<B-TEST> assessment<I-TEST> is<O> non-focal<B-
    PROBLEM> .<O> Overall<O> ,<O> there<O> are<O> no<O> evident<O> physical<O>
    findings<O> to<O> explain<O> the<O> fever<B-PROBLEM> apart<O> from<O> the<O>
    stated<O> temperature<B-MEASUREMENT> elevation<I-MEASUREMENT> .<O>

Do not give any other explianation of additional text. Only output exactly in the
    format specified above, make sure break up any forms of puntuations and that
    you start your BIO tags leading up to words too, for example "shortness of
    breath" would be fully tagged as "shortness<B-PROBLEM> of<I-PROBLEM> breath <I-
    PROBLEM>" not only tagging "breath". Also if you have fractions make sure you
    divide it properly like "98%" would be "98<B-MEASUREMENT> %<I-MEASUREMENT>" and
     "120/80" would be "120<B-MEASUREMENT> /<I-MEASUREMENT> 80<I-MEASUREMENT>".
```

Contest for Information Sharing

As part of this course, we may share selected project materials (e.g., reports, presentation slides, and presentation recordings) on the course webpage as learning resources for future students. Additionally, we may use anonymized project information for internal statistical analysis of course outcomes. Please indicate your preferences below. *Your choices will not affect your grade in any way.*

Consent for Sharing Project Materials

Please keep the item you wish and remove the other one.

- All group members consent to allow the project materials (report, slides, and presentation recording) to be shared on the course page for future students.

Consent for Use of Project Information in Statistical Analysis

Please keep the item you wish and remove the other one.

- All group members consent to allow anonymized information about the project (e.g., topic, methods, outcomes, grades) to be used by the instructor for statistical analysis and course improvement.

Optional Comments

Please list any specific conditions or comments here.

Group Identification

- Group number: 13
- Names of group members: Jason Khuu, Weihan Xie
- Signature of of group members: Jason Khuu, , Weihan Xie
- Date: 2025-12-10